DB2 and PHP

New features, best practices on IBM i

Alan Seiden Consulting

alanseiden.com

Alan’s PHP on IBM i focus

- Consultant to innovative IBM i and PHP users
- PHP project leader, Zend/IBM Toolkit
- Contributor, Zend Framework DB2 enhancements
- Award-winning developer
- Authority, web performance on IBM i

Founder, Club Seiden

club.alanseiden.com

Contact information

Alan Seiden

alan@alanseiden.com

201-447-2437

alanseiden.com

twitter: @alanseiden

Today’s discussion will include:

- **Uniqueness of DB2 for IBM i**
- **Which DB2-enabled middleware to use with PHP**
- **Securing your SQL with prepared queries**
- **Connection options for speed and reliability**
  - Persistent connections
  - Library lists
- **What’s NEW in ibm_db2**
- **Connecting from “off the box”**
- **Many other tips**

Why learn DB2 best practices?

- **As chief database on IBM i, DB2 runs these:**
  - Most transaction processing systems
  - Stored procedures
  - “XMLSERVICE” Toolkit
    - Accessible with db2 stored procedures from PHP

- **DB2 knowledge will help you:**
  - Maximize speed
  - Reduce CPU usage
  - Maximize reliability
    - Avoid unexpected locking and other operational problems

Heart of IBM i is DB2

- **DB2 built in**
  - Transaction processing workhorse
  - Database implemented below operating system level!
    - IBM i's "Machine Interface (MI)" between OS and hardware
  - Journaling, auditing, commitment control very commonly used
  - Never corrupted
    - Doesn't lose data even if knock out power plug

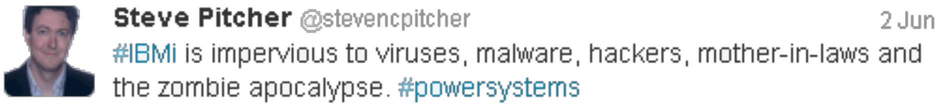
- **Database often taken for granted**
  - So self-managing, DBAs are rare

Reliable

Nondisruptive business growth

- **Scales vertically**
  - One system can handle large and diverse workloads
    - Total Cost of Ownership (TCO), including reduced operator costs, is said to be competitive or cheaper than assembling server farms
  - Can activate additional processors without restarting system

- **Dependable**

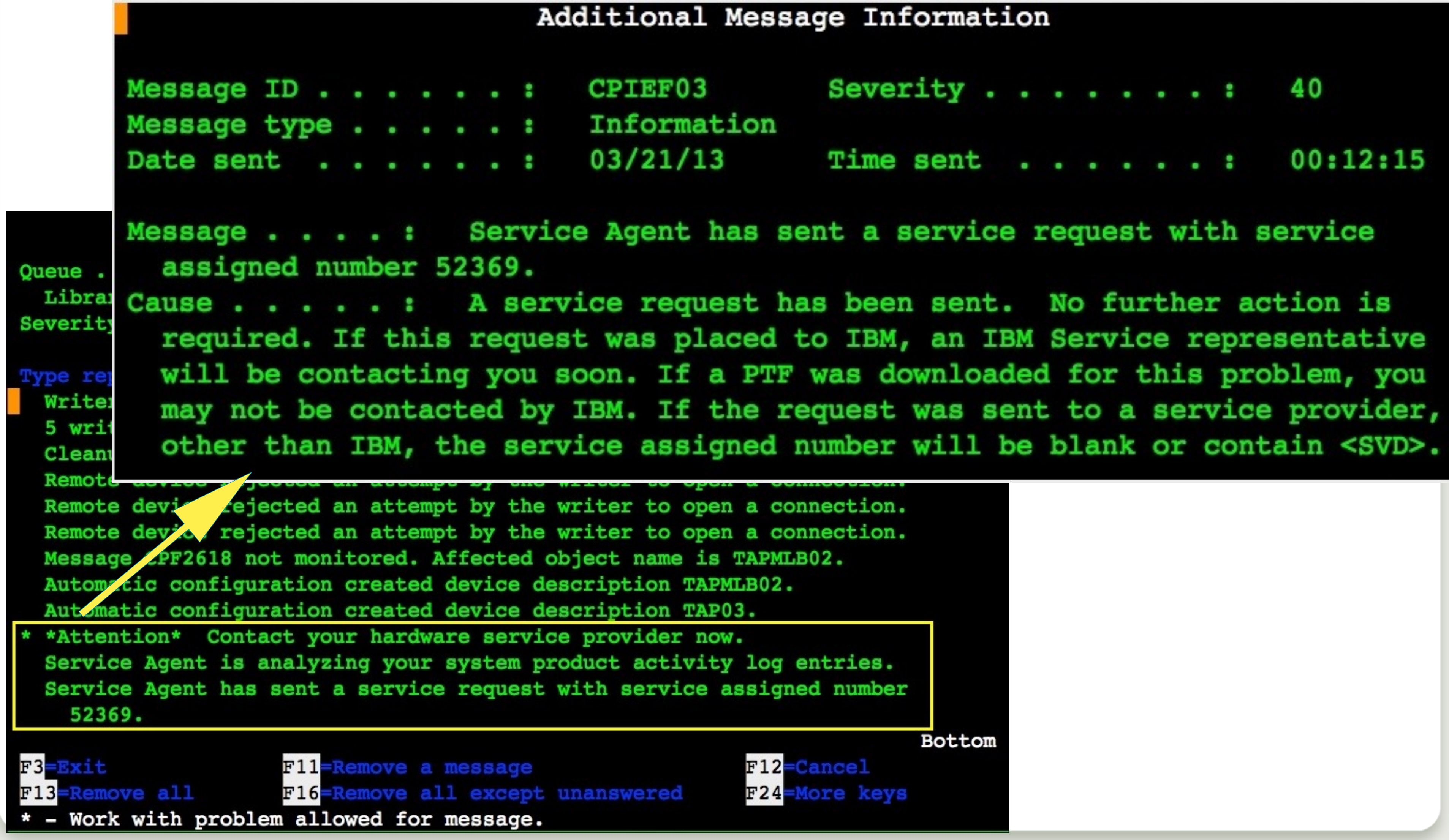


Steve Pitcher @stevencpitcher 2 Jun

#IBMi is impervious to viruses, malware, hackers, mother-in-laws and the zombie apocalypse. #powersystems

  - Resistant to viruses
    - Object-based system since the 1970s
  - Journaling, commitment control, replication, high availability
  - Security features galore
  - Keeps on running
    - You will sleep soundly at night

IBM i can “phone home”

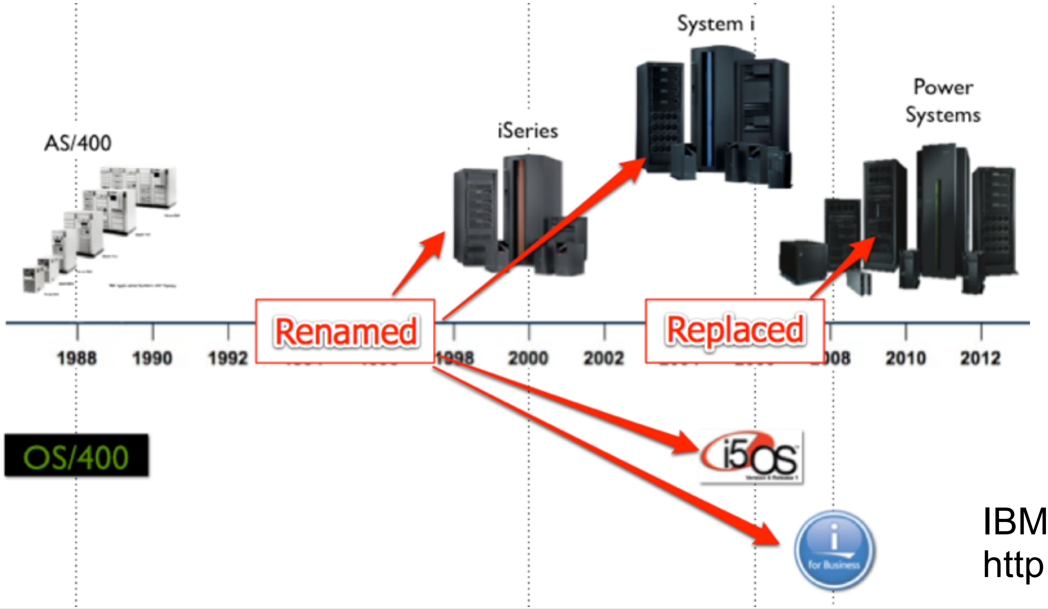


```
                       Additional Message Information

Message ID . . . . . . :   CPIEF03       Severity . . . . . . . :   40
Message type . . . . . :   Information
Date sent  . . . . . . :   03/21/13      Time sent  . . . . . . :   00:12:15

Message . . . . :   Service Agent has sent a service request with service
  assigned number 52369.
Cause . . . . . :   A service request has been sent.  No further action is
  required. If this request was placed to IBM, an IBM Service representative
  will be contacting you soon. If a PTF was downloaded for this problem, you
  may not be contacted by IBM. If the request was sent to a service provider,
  other than IBM, the service assigned number will be blank or contain <SVD>.
```

```
Queue .
  Librar
Severity

Type rep
  Write
  5 wri
  Clean
  Remote
  Remote device rejected an attempt by the writer to open a connection.
  Remote device rejected an attempt by the writer to open a connection.
  Message CPF2618 not monitored. Affected object name is TAPMLB02.
  Automatic configuration created device description TAPMLB02.
  Automatic configuration created device description TAP03.
* *Attention*  Contact your hardware service provider now.
  Service Agent is analyzing your system product activity log entries.
  Service Agent has sent a service request with service assigned number
    52369.
                                                                      Bottom
F3=Exit                 F11=Remove a message                F12=Cancel
F13=Remove all          F16=Remove all except unanswered    F24=More keys
* - Work with problem allowed for message.
```

Data and programs last forever

- **IBM has been dedicated to legacy app longevity and data longevity**
  - RPG (and occasionally COBOL) running for 30-40 years
  - DB2 data evolving 30-40 years
  - Middleware insulates applications from hardware changes



IBM i Heritage chart from Trevor Perry
http://blog.angustheitchap.com/?p=415

Long-serving back ends, new front ends

- **What does this mean to you?**
  - RPG and DB2, mature and evolving for years, can be part of your data model, accessed by PHP
  - Create web GUI interfaces and web services around these venerable resources
    - Business logic is encapsulated in RPG/COBOL/DB2
    - You can keep your hands somewhat clean of business details

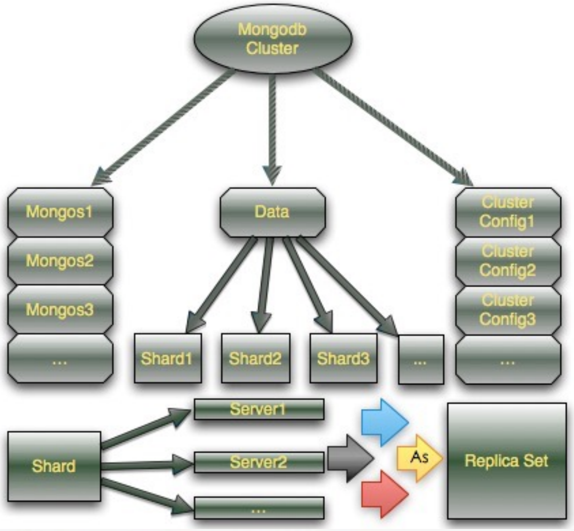
DB2 on IBM i is…

- **Fully integrated business database**
  - Coded at the kernel level (below the OS)
    - Other database systems are .exe files
  - User profiles = real IBM i user profiles
    - Security, logging, auditing consistent throughout the system
    - Other databases have “pretend” users whose security is enforced only by database code
- Always present and available
  - No daemon that can be ended or hacked
  - “Who shut down the database?” Not on IBM i!

DB2 on IBM i is…

- **Indestructible, low maintenance (cost-effective)**
  - Data corruption almost unheard of
  - 30+ years of work for reliability and throughput
  - DBA-less operation. Scale vertically. Add disk and go!

- DB2 on i is **not** this:



Disclaimer: Not singling out other databases as “bad.” A NoSQL database such as Mongo prizes flexibility more than management ease

MongoDB @MongoDB
Sharding Pitfalls Part III: Chunk Balancing and Collection Limits buff.ly/1rgktMc

Database terms

- **Modern and traditional IBM i terminology coexist**

Modern term	Traditional term or phrase
Schema	Library
Table	File or Physical File
Index	Logical File*
Row	Record
Column	Field

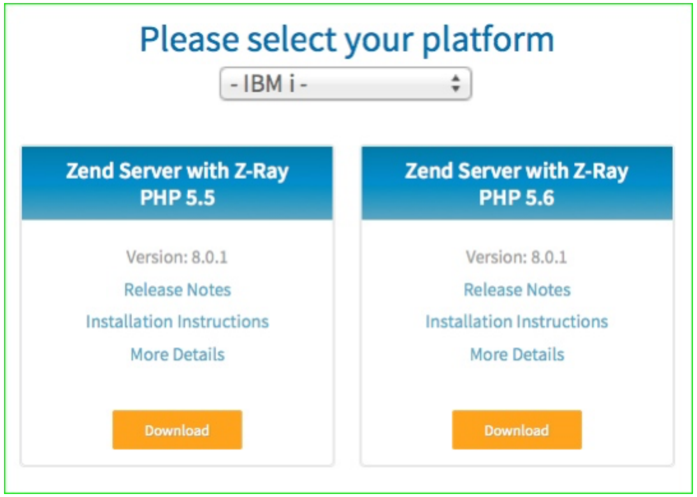
* a logical file resembles an "index + view"

Modern views, triggers, etc. are supported by DB2.

Prerequsites for DB2 with PHP

Zend Server for IBM i

- **Download Zend Server 8.x**
  - http://www.zend.com/en/products/server/downloads-ibmi
    - Easy upgrade from 6.x
    - Includes ibm_db2 version 1.9.6-sg25 with many updates



Please select your platform

- IBM i -

Zend Server with Z-Ray PHP 5.5

Version: 8.0.1

Release Notes

Installation Instructions

More Details

Download

Zend Server with Z-Ray PHP 5.6

Version: 8.0.1

Release Notes

Installation Instructions

More Details

Download

- **Editions**
  - Basic (free), Professional, Enterprise
  - http://www.zend.com/en/products/server/editions
  - Same download, different license

DB2 is an exciting growth zone for IBM i

- **Behooves you to stay on top of it**
- **Example of update:**
  - http://www.mcpressonline.com/ibm-i-os/400-i5/os/step-right-up-and-hear-about-db2-and-tr9.html
    - Regular expressions in WHERE clause
    - System info available via SQL…joblog, liblist, more

DB2 drivers

Choice of middleware to access DB2

- **Zend Server includes at least three such extensions:**

- **odbc**
  - Less functionality than the others (generic)
  - “Free” connections from other platforms

- **IBM_PDO**
  - PDO = PHP Data Objects
  - Generic DB2. No “/” separator or library lists (yet)

- **ibm_db2**
  - Provides IBM i-specific features such as library list support

ibm_db2 documentation

- **Manual page**
  - http://php.net/ibm_db2
- **Source code and additional documentation at the “PECL” PHP extension repository**
  - http://pecl.php.net/package/ibm_db2
  - Read the “C” source sometime—it’s educational
- **We will examine ibm_db2 in detail today**

Connect to DB2

Connect with db2_connect()

- **db2_connect() creates a database connection**
  - Accepts four parameters that you should master
    - Three string parameters
    - One array of optional options

- **resource db2_connect (string $database,**
  string $username,
  string $password
  [, array $options])

- **db2_pconnect() is similar**
  - pconnect creates persistent connections (more on that later)

db2_connect() string parameters

- **$database**
  - Leave blank ('') for default local database
  - Use a db name from WRKRDBDIRE for a choice of databases
    - Database name can be *LOCAL or that of an LPAR, IASP (Independent auxiliary storage pool), or another machine
- **$username**
  - Leave blank ('') for default Apache user (QTMHHTTP)
    - Not recommended
  - Use any valid user profile to associate queries with that user
- **$password**
  - Leave blank ('') if $database and $username were blank
  - Otherwise, provide password corresponding to $username

db2_connect() basic examples

- **Empty params**
  - `$conn = db2_connect('', '', '');`
  - Connects to local database with web user QTMHHTTP
  - Not recommended: may be disallowed in future release

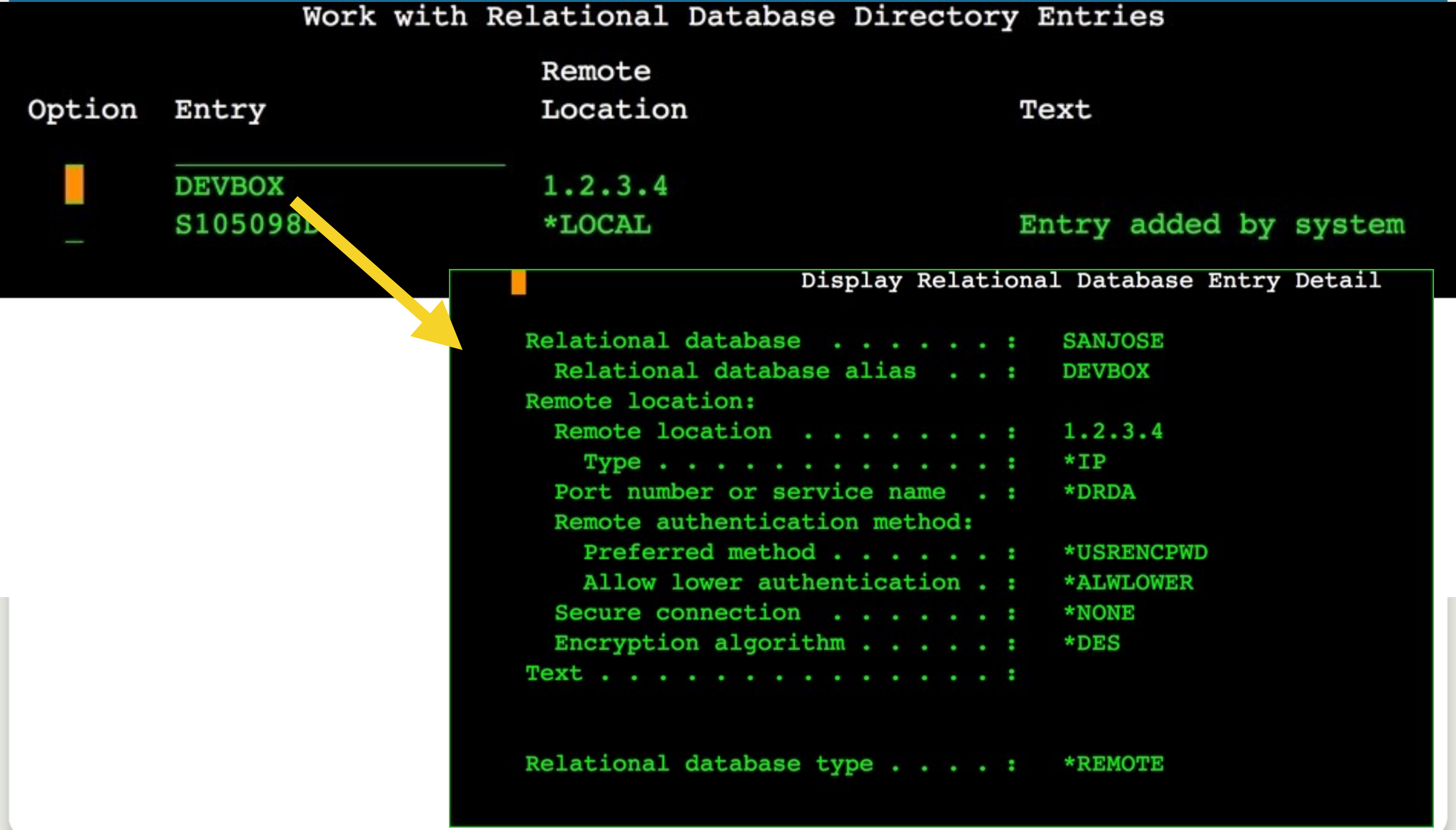
- **Better: use specific values**
  - `$conn = db2_connect('MYDB', 'MYUSER', 'MYPASS');`
  - Connects to MYDB database (must be configured in WRKRDBDIRE) with user MYUSER

How to connect to remote DB or iASP

- Connect to another partition, server, or iASP as if local
- Useful for testing PHP scripts against databases of multiple partitions (dev/test/production)

- Example: remote server is at IP 1.2.3.4 and has database named SANJOSE
- We will refer to it on our box with alias “DEVBOX”
- Use WRKRDBDIRE (Work with Relational Database Directory Entries) to create local alias to remote database

WRKRDBDIRE to set up alias



```
              Work with Relational Database Directory Entries

                          Remote
Option  Entry             Location                    Text

        DEVBOX            1.2.3.4
        S105098D          *LOCAL                      Entry added by system
```

```
                         Display Relational Database Entry Detail

   Relational database  . . . . . . . :   SANJOSE
     Relational database alias  . . . :   DEVBOX
   Remote location:
     Remote location  . . . . . . . . :   1.2.3.4
       Type . . . . . . . . . . . . . :   *IP
     Port number or service name  . . :   *DRDA
     Remote authentication method:
       Preferred method . . . . . . . :   *USRENCPWD
       Allow lower authentication . . :   *ALWLOWER
     Secure connection  . . . . . . . :   *NONE
     Encryption algorithm . . . . . . :   *DES
   Text . . . . . . . . . . . . . . . :


   Relational database type . . . . . :   *REMOTE
```

Now we can access “DEVBOX” database

- User and password must be correct for the remote database

```
$db = db2_connect('DEVBOX', 'DEVUSER', 'DEVPWD');
```

Fourth parameter (array $options)

- **Optional array to fine-tune the connection**
- **Below are choices that are most relevant for IBM i**
- **Details as we go**
  - i5_lib
    - Set a single default library
  - i5_naming
    - Choose “system” or SQL naming
  - i5_libl
    - Set a library **list** (be sure to set i5_naming on)
  - i5_commit
    - Commitment control options
  - autocommit
    - DB2_AUTOCOMMIT_ON (default) or _OFF

i5_lib

- **Specify one library as default**
  - 'i5_lib'=>'MYLIB'
- **Any unqualified files/tables will be assumed to use this library**

i5_naming for library lists

- **DB2_I5_NAMING_ON**
  - A constant equal to 1 that turns on “system naming” mode
  - Table files qualified using the slash (/) delimiter
  - Unqualified files are resolved using the library list for the job

- **DB2_I5_NAMING_OFF**
  - A constant equal to 0 (default) that enables “SQL naming” mode
  - Table files qualified using the period (.) delimiter
  - Unqualified files are resolved using either the default library (i5_lib) or the user profile name specified on db2_connect() (could be QTMHHTTP)
    - Message to watch for:
      MYTABLE in YOURNAME type *FILE not found.
      SQLCODE=-204”

User profile strategy

Three techniques (1 of 2)

- Small number of "system" user profiles. One user per library list. When "real" user signs in, authenticate against a database, LDAP, etc. Not against user profile list

Three techniques (2 of 3)

- db2_connect with real user profile
  - user can get disabled when wrong password is entered
  - revealing an actual user profile to end users (OK for internal. security problem if external)
  - benefit: authority, program logic, journal entries will work with "real" IBM i user profile

- db2_pconnect works the same except that the jobs stay semi-active, visible to operator

Three techniques (3 of 3)

- db2_pconnect with generic "system" profile but then switch to "real" profile after authentication
  - fast connection
  - limits number of active jobs
  - performance cost of the "switch"
  - security concerns about mixed generic/specific user profile attributes/authority in a job
  - need to "switch back" afterward

Library list jubilee (3 ways to set)

Most efficient way to set library list

- **Let IBM i set library list based on user profile**
  - No extra program calls required
- **Use a user profile that has an "initial library list" in its job description (JOBD)**
- **Specify the user profile and i5_naming=ON**

  -
    ```
    $conn = db2_connect('*LOCAL', 'LIBLUSER', 'PASS',
    array('i5_naming' => DB2_I5_NAMING_ON));
    ```

“JOBD” technique in detail

Create jobd:

```
CRTJOBD  JOBD(QGPL/APPROD)
         INLLIBL(LIB1 LIB2 LIB3 QGPL)
```

Set jobd in user profile (or create new profile):

```
CHGUSRPRF  USRPRF(APPRODUSR)
           JOBD(QGPL/APPROD)
```

- **In PHP, specify the user profile and i5_naming=ON**

  -
    ```
    $conn = db2_connect('*LOCAL', 'LIBLUSER', ‘PASS’,
    array('i5_naming' => DB2_I5_NAMING_ON));
    ```

Middle ground: i5_libl

- $options array accepts 'i5_libl'
    - i5_libl is a space-delimited library list

```
'i5_libl'=>'MYLIB YOURLIB ANYLIB’
```

- Causes ibm_db2 to run CHGLIBL at the middleware level
- Example:

```
$conn = db2_connect('*LOCAL' ,'MYUSER', 'MYPASS',
                  array('i5_naming' => DB2_I5_NAMING_ON,
                        'i5_libl'   => 'MYLIB1 MYLIB2’));
```

- Note: with persistent connections, CHGLIBL only run the first time in, so be sure to distinguish different jobs by user profile

More work but most flexible

- **If maximum flexibility required:**
  - Run a command or program that sets up library list
  - Use CHGLIBL instead of ADDLIBLE
    - ADDLIBLE could generate “already in library list” error, causing toolkit to retrieve joblog (slow)

```
$db = db2_connect('*LOCAL', 'MYUSER', 'MYPASS',
            array('i5_naming' => DB2_I5_NAMING_ON));
// connect to toolkit using existing DB2 conn
$tkitConn = ToolkitService::getInstance($db,
            DB2_I5_NAMING_ON);
// toolkit will share job with DB2
$tkitConn->setOptions(array('stateless' => true));

$tkitConn->CLCommand('CHGLIBL LIBL(NXSRTQA QGPL
            QTEMP)');
```

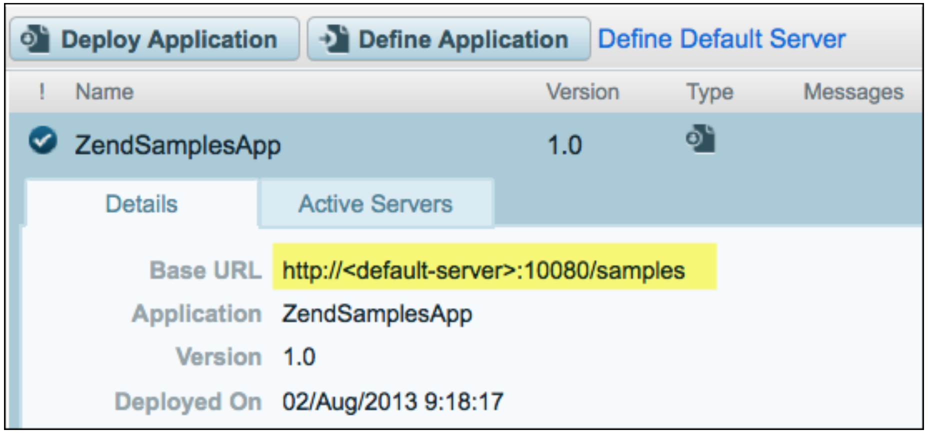
db2_connect() example with $options

```
$database = 'MYDB';
$user = 'MYUSER';
$password = 'MYPASS';

$options = array('i5_naming'  => DB2_I5_NAMING_ON,
                 'i5_libl'    => 'MYLIB1 MYLIB2'
);
$conn = db2_connect($database, $user, $password, $options);

if ($conn) {
    echo "Connection succeeded.";
} else {
    echo "Connection failed.";
}
// MYTABLE will be found, if in library MYLIB1 or MYLIB2
$stmt=db2_exec($conn,"SELECT * FROM MYTABLE");
```

Included sample script

Included with Zend Server

“SQL Access” sample script illustrates several techniques from this talk



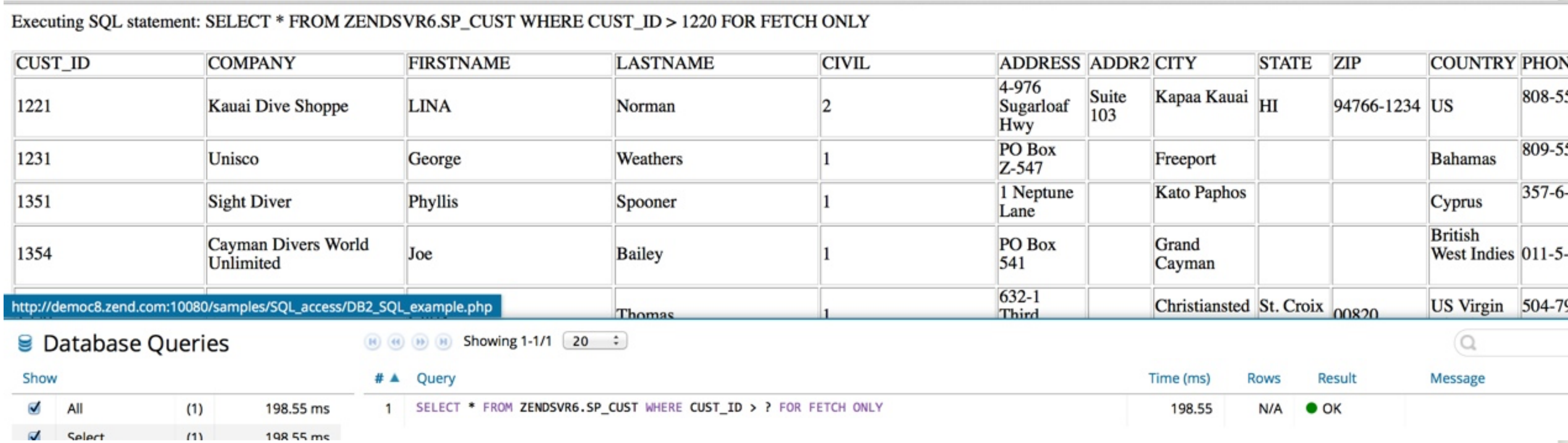
ZendSamplesApp 1.0

Base URL http://<default-server>:10080/samples

Application ZendSamplesApp

Version 1.0

Deployed On 02/Aug/2013 9:18:17



Executing SQL statement: SELECT * FROM ZENDSVR6.SP_CUST WHERE CUST_ID > 1220 FOR FETCH ONLY

CUST_ID	COMPANY	FIRSTNAME	LASTNAME	CIVIL	ADDRESS	ADDR2	CITY	STATE	ZIP	COUNTRY	PHON
1221	Kauai Dive Shoppe	LINA	Norman	2	4-976 Sugarloaf Hwy	Suite 103	Kapaa Kauai	HI	94766-1234	US	808-55
1231	Unisco	George	Weathers	1	PO Box Z-547		Freeport			Bahamas	809-55
1351	Sight Diver	Phyllis	Spooner	1	1 Neptune Lane		Kato Paphos			Cyprus	357-6-
1354	Cayman Divers World Unlimited	Joe	Bailey	1	PO Box 541		Grand Cayman			British West Indies	011-5-
			Thomas	1	632-1 Third		Christiansted	St. Croix	00820	US Virgin	504-79

http://democ8.zend.com:10080/samples/SQL_access/DB2_SQL_example.php

Database Queries

Showing 1-1/1 20

Show		
☑ All	(1)	198.55 ms
☑ Select	(1)	198.55 ms

# ▲	Query	Time (ms)	Rows	Result	Message
1	SELECT * FROM ZENDSVR6.SP_CUST WHERE CUST_ID > ? FOR FETCH ONLY	198.55	N/A	● OK	

Snippet of sample script

```
/* Construct the SQL statement */
$sql = "SELECT * FROM ZENDSVR.SP_CUST WHERE CUST_ID > ? FOR FETCH ONLY";

/* Prepare, bind and execute the DB2 SQL statement */
$stmt= db2_prepare($conn_resource, $sql);
$lower_limit = 1220; //from the CUST_ID value
$flds = db2_num_fields($stmt);
if($flds > 0 )
{
//show Table Header (analyze result set)
      echo "<table border=1>";
      echo "<tr>";
      for($i=0; $i<$flds; $i++)
      {
            echo '<td width="20%">';
            $name = db2_field_name($stmt, $i);
            echo $name;
            echo "</td>";
      }

      echo "</tr>";

//Execute statement , uses a binding of parameters
db2_bind_param($stmt, 1, "lower_limit", DB2_PARAM_IN);
$result = db2_execute($stmt);
```

Commitment control

Assure data integrity

- Commitment control allows "all or none" logic when running multiple update/insert/delete queries
- If one query fails, roll back previous related queries
- Example: If a detail record update fails, roll back the header update
- Requires that journaling be enabled on files/tables being written to

Use commitment control for data integrity

- **Example: “all or none” for two INSERTS**
  - with ibm_db2.i5_allow_commit = 1
  - and autocommit = DB2_AUTOCOMMIT_OFF

```
$conn = db2_pconnect('', '', '', array('autocommit'=>DB2_AUTOCOMMIT_OFF));
$stmt=db2_prepare($conn,"INSERT INTO MYTABLE (IDNUM, NAME) VALUES(?, ?)");

$result1 = db2_execute($stmt, array(1, 'jane')); // should insert OK
$result2 = db2_execute($stmt, array('x', 'bob')); // not numeric!

// check if both INSERTs succeeded
if ($result1 && $result2) {
  // Success. Commit both inserts
  db2_commit($conn);
} else {
  // *** Error with one of the inserts; roll them both back ***
  db2_rollback($conn);
}
// Neither record will be in the table. We rolled back.
```

i5_commit

- **i5_commit**
  - Options for isolation level
  - Gives you fine-grained control (and ability to turn off altogether)
  - Before 1.9.6.-sg25, must also enable commitment control system-wide
    - ibm_db2.i5_allow_commit = 1 in INI file
  - Choices:
    - *DB2_I5_TXN_NO_COMMIT* – turns off commitment control for this connection
    - *DB2_I5_TXN_READ_UNCOMMITTED*
    - *DB2_I5_TXN_READ_COMMITTED*
    - *DB2_I5_TXN_REPEATABLE_READ*
    - *DB2_I5_TXN_SERIALIZABLE*

autocommit

- **DB2_AUTOCOMMIT_ON**
  - A constant equal to 1 (default)
  - Turns autocommit on
    - End of script causes commit
  - Only relevant when commitment control is used
  - Convenient: insert/update/delete will work without db2_commit()

- **DB2_AUTOCOMMIT_OFF**
  - A constant equal to 0
  - Turns autocommit off
  - Only relevant when commitment control is used
  - Provides flexibility to ensure data integrity in multi-step transactions by using db2_commit()/db2_rollback() around groups of insert/update/delete queries

Commitment control tips

- **In php.ini:**
  - To use commitment control before 1.9.6-sg25, set ibm_db2.i5_allow_commit = 1
- **In db2_connect() option array:**
  - Modify default settings with ‘i5_commit’ option
  - Choose ‘autocommit’ on or off
- **Turn on journaling for schemas (libraries)**
  - Already on if schema created via “CREATE SCHEMA” (SQL/DDL)
  - Extra step needed for libraries created via CRTLIB
    - Start Journal Library (STRJRNLIB, v6.1+) makes a library a journaled object. Any objects eligible to be journaled that are added to the library can be automatically journaled
    - http://www.redbooks.ibm.com/abstracts/tips0662.html
    - http://kb.zend.com/index.php?View=entry&EntryID=235

More about db2_connect, db2_pconnect

- **Manual pages**
  - http://www.php.net/manual/en/function.db2-connect.php
  - http://www.php.net/manual/en/function.db2-pconnect.php
  - http://www.php.net/manual/en/features.persistent-connections.php

Prepare queries

“Prepared” = safe and fast

- **Prepared queries help in several ways**
  - Eliminate errors due to un-escaped single quotes
  - Protect your data from SQL Injection attacks
  - Speed up repeated queries

- **They are also known as prepared statements**

- **Here’s an example of the mischief they prevent**

Apostrophes confuse query parsers

```
// mysite.com?name=whatever
$name = $_GET['name'];
$sql = "select custno from custfile
 where name = '$name' and status = 'ACTIVE' ";
```

- Do you see any potential problems?

- What if the name is “O’Shea” ? Error!

```
$sql = "select custno from custfile
 where name = 'O'Shea' and status = 'ACTIVE' ";
```

- Single quotes confuse query parser when they serve two purposes
  - Used as apostrophe in data
  - Delimiter of string literals in the SQL syntax

Malicious users can try “SQL Injection”

```
// mysite.com?name=whatever
$name = $_GET['name'];
$sql = "select custno from custfile
 where name = '$name' and status = 'ACTIVE' ";
```

- What if the name is the weird-looking “`x' OR 1=1--`”

(That is, a user typed: mysite.com?name=x' OR 1=1--)

```
 $sql = "select custno from custfile
 where name = 'x' OR 1=1--' and status = 'ACTIVE' ";
```

- **Every record in the table will be selected**
  - OR 1=1 will always be true
  - -- turns subsequent ‘where’ criteria into a comment (ignored!)

Safeguard data with prepared queries

```
// mysite.com?name=whatever
$name = $_GET['name'];
$sql = "select custno from custfile
 where name = ? and status = 'ACTIVE' ";
```

- Represent parameters with question marks (?) instead of literal values
- It’s fine to retain hard-coded values in the query
  - Such as ‘ACTIVE’ in the example above
- Supply parameters in an array
  - $params = array(“O'Shea”);

- Full example on next slide

db2_prepare() with db2_execute()

```
$name = $_GET['name'];
$conn = db2_connect('','','');
$sql = "select custno from custfile
 where name = ? and status = 'ACTIVE' ";
$params = array($name); // can be "O'Shea" for all we care
$stmt = db2_prepare($conn, $sql);
if ($stmt) { // prepared OK
    $result = db2_execute($stmt, $params);
    if ($result) { // ran query OK with parameters
        while ($row = db2_fetch_assoc($stmt)) {
            echo "$row['custno']\n";
        }
    }
}
```

Ordinary db2_exec() re-calcs plan

- Ex. of non-prepared SQL repeated with different params

```
$values = array('acme', 'shoprite', 'stop n shop');
foreach ($values as $value) {
    $sql = "select custno from custfile
            where name = '$value' and status = 'ACTIVE' ";
    // query gets re-optimized in each iteration
    $stmt = db2_exec($conn, $sql);
    if ($stmt) { // do something with $stmt }
}
```

- The query plan will re-optimize on each db2_exec() because a new SQL string was supplied each time
- OK for one-off queries but not when repeated

Prepared statement allows re-use of plan

- Ex. of prepared SQL; execution with different params

```
// prepare the query ONCE
$sql = "select custno from custfile
            where name = ? and status = 'ACTIVE' ";
$stmt = db2_prepare($conn, $sql);
// now execute with values only
$values = array('acme', 'shoprite', 'stop n shop');
foreach ($values as $value) {
    $result = db2_execute($stmt, array($value));
    if $result {// do something with $stmt }
}
```

- The query plan is calculated ONCE and reused with each db2_execute(), saving time and CPU

Prepared statements/queries are best

- **Replace db2_exec() with db2_prepare() and db2_execute()**

- **Benefits**
  - Queries will run as intended, with fewer surprises
  - Protection from a common form of hacking (SQL injection)
  - Performance will improve for repeated queries

Check result

Connections return a resource or false

```
$conn = db2_connect("*LOCAL", "MYUSER", "BADPASS");

    // test for false
    if (!$conn) {
        echo "Connection failed. SQL Err: ";
        echo db2_conn_error() . "<br>";
        echo db2_conn_errormsg();

        die();

    } else {
        // use the connection....
    }
```

An incorrect password will generate this output:

```
Connection failed. SQL Err: 08001
Authorization failure on distributed database
connection attempt. SQLCODE=-30082
```

Prepare/execute: resource or false

```
$sql = "SELECT * FROM BADLIB.SP_CUST WHERE CUST_ID > ?";

$stmt= db2_prepare($conn, $sql);

if (!$stmt) {
    echo 'The db2 prepare failed. ';
    echo 'SQLSTATE value: ' . db2_stmt_error() . '<BR>';
    echo ' Message: ' .   db2_stmt_errormsg();
}
```

The error code and message might resemble:

```
SQLSTATE value: 42704
Message: SP_CUST in BADLIB type *FILE not found.
SQLCODE=-204
```

Functions to get error codes/messages

• **db2_conn_error()**	connection error code
• **db2_conn_errormsg()**	connection error text
• **db2_stmt_error()**	prepare/execute error code
• **db2_stmt_errormsg()**	prepare/execute error text

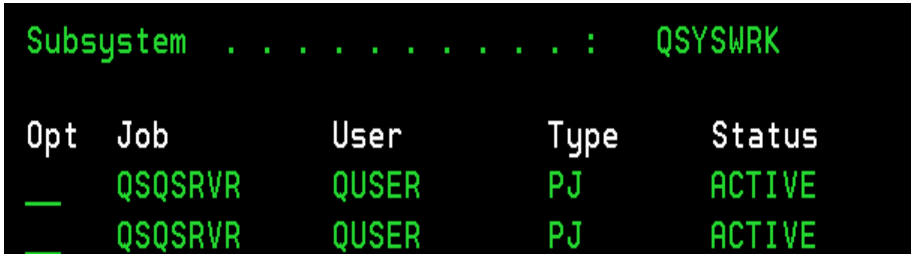
- Tip: a statement error could also occur while connecting. How? if ‘i5_libl’ is specified with incorrect libraries, the internal CHGLIBL fails and you get a statement error.

Where DB2 runs

Server Mode

DB2 queries run in separate prestart jobs

- **Advantages**
  - Can choose user profile
  - Can use persistent connections (fast!)
  - Connect to multiple RDBs (databases) concurrently from PHP
- **Disadvantages**
  - Potential delays to spawn jobs, if non-persistent
  - Creates *SHRRD pseudo-locks that may require attention (more on this later)



```
Subsystem  . . . . . . . . . . :   QSYSWRK

Opt  Job         User        Type     Status
 __  QSQSRVR     QUSER       PJ       ACTIVE
 __  QSQSRVR     QUSER       PJ       ACTIVE
```

Server Mode configuration

- **Prestart jobs named QSQSRVR run in QSYSWRK**
- **Or, if remote DRDA, QRWTSRVR in QUSRWRK**
- **Configurable pool of jobs**

```
CHGPJE SBSD(QSYS/QSYSWRK) PGM(QSYS/QSQSRVR)
  STRJOBS(*YES) INLJOBS(xx) THRESHOLD(xx)
  ADLJOBS(xx) MAXUSE(xx or *NOMAX)
```

- **More information about server mode**
  - http://www.redbooks.ibm.com/abstracts/tips0658.html
  - http://www.mcpressonline.com/tips-techniques/database/techtip-grab-control-of-the-db2-qsqsrvr-jobs.html
  - http://www.mcpressonline.com/database/db2/finding-sql-server-mode-connecting-jobs.html

Set up DB2/i
in
Zend Framework

IBM i configuration change

In application.config.php, add __DIR__

```
 'config_glob_paths' => array(
            __DIR__ . '/autoload/{,*.}
{global,local}.php',
 ),
```

This is because IBM i normally starts searching at the IFS root. Must tell it to start at __DIR__.

Set up IBM i DB2 connection

In local.php (or combination global and local)

```
return array(
    'db' => array(
        'driver' => 'IbmDb2',
        'database' => ‘*LOCAL',
        'username' => 'MYUSER',
        'password' => 'MYPASS',
        ‘persistent' => true, // true or false. Update your ZF2 for this
        'driver_options' => array(
            'autocommit' => 'DB2_AUTOCOMMIT_OFF',
            'i5_lib' => ‘ZENDSVR6', // for example. Could also specify i5_libl
        ),
    ),
    'service_manager' => array(
        'factories' => array(
            'db-adapter' => 'Zend\Db\Adapter\AdapterServiceFactory',
        ),
    ),
);
```

Pagination without LIMIT

Email from a PHP fan

“I’ve run into a little problem with DB2. I’m used to being able to use the **LIMIT option in MySQL** to set a range of records I want to view, perhaps the 20-29 records for instance.

“Could you tell me if there is or is not a way to do that on IBM i? My boss and I are beginning to think that such an option does not exist for use on the AS400 IBM i.”

- Yes, it can be done via a choice of two techniques
- Useful for pagination (page-at-a-time logic)
- LIMIT and OFFSET are non-standard, not in DB2

1. ibm_db2 middleware technique

```
// specify DB2_SCROLLABLE in db2_exec or db2_execute
$startingRow = 20;
$endingRow = 29;
$stmt = db2_execute($stmt, array('cursor' =>
DB2_SCROLLABLE));
$currentRow = $startingRow;
while (($currentRow <= $endingRow) &&
           $row = db2_fetch_array($stmt, $currentRow)) {
    print "$row[0]\n";
    $currentRow++;
}
```

Note additional param for row number:

array db2_fetch_array (resource $stmt [, int $row_number = -1])

Disadvantage: middleware-dependent (only works with ibm_db2)

2. SQL technique

- DB2’s row_number() and over() functions can select records by number in a recordset
- This simulates LIMIT and OFFSET

- Example coming up

Record range selection in DB2

Start with this:

```
$queryString = "SELECT CUST_ID, COMPANY FROM SP_CUST
order by CUST_ID”
```

Use row_number() and over() to limit the record selection:

```
$queryString = "SELECT CUST_ID, COMPANY FROM
  (select row_number() over (order by CUST_ID) as rowid,
CUST_ID, COMPANY
   from SP_CUST) as t
   where t.rowid between 20 and 29";
```

Or let Zend Framework handle it

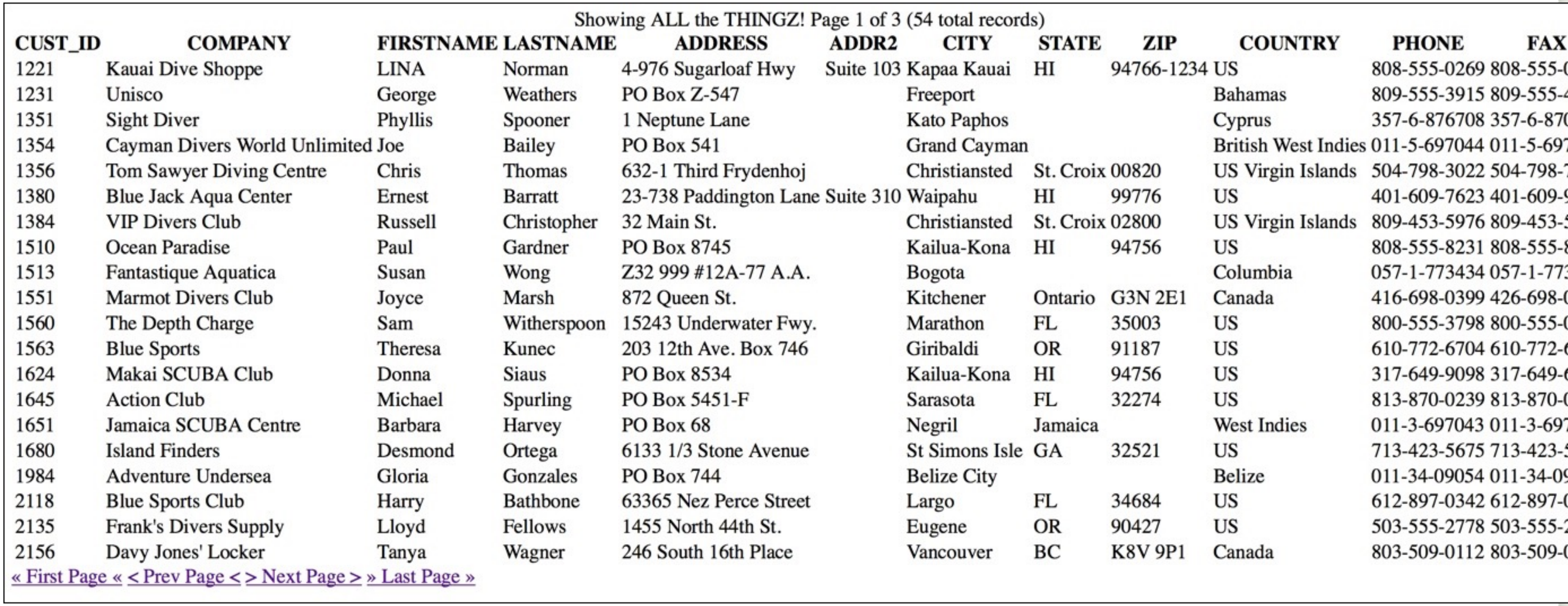
- **Use ZF's Zend Db component from regular PHP**
- **Builds SQL, hiding complexity of LIMIT/OFFSET implementation, with cross-database code**
- **Full script:** https://github.com/alanseiden/Code-Examples/blob/master/IBMi/DB2/ZF2LimitOffset.php

```
// works with v2.3.2 of ZF2.
$sql = new Sql($adapter);
$select = $sql->select();
$select->from('SP_CUST')
       ->where('CUST_ID > 1220')
       ->order('CUST_ID ASC')
       ->limit(10)
       ->offset(20);
```

Sample DB2 on i pagination script

coded by Zend’s Clark Everetts

https://github.com/clarkphp/Code-Examples/tree/master/IBMi/pagination



Showing ALL the THINGZ! Page 1 of 3 (54 total records)

CUST_ID	COMPANY	FIRSTNAME	LASTNAME	ADDRESS	ADDR2	CITY	STATE	ZIP	COUNTRY	PHONE	FAX
1221	Kauai Dive Shoppe	LINA	Norman	4-976 Sugarloaf Hwy	Suite 103	Kapaa Kauai	HI	94766-1234	US	808-555-0269	808-555-0
1231	Unisco	George	Weathers	PO Box Z-547		Freeport			Bahamas	809-555-3915	809-555-4
1351	Sight Diver	Phyllis	Spooner	1 Neptune Lane		Kato Paphos			Cyprus	357-6-876708	357-6-870
1354	Cayman Divers World Unlimited	Joe	Bailey	PO Box 541		Grand Cayman			British West Indies	011-5-697044	011-5-697
1356	Tom Sawyer Diving Centre	Chris	Thomas	632-1 Third Frydenhoj		Christiansted	St. Croix	00820	US Virgin Islands	504-798-3022	504-798-7
1380	Blue Jack Aqua Center	Ernest	Barratt	23-738 Paddington Lane	Suite 310	Waipahu	HI	99776	US	401-609-7623	401-609-9
1384	VIP Divers Club	Russell	Christopher	32 Main St.		Christiansted	St. Croix	02800	US Virgin Islands	809-453-5976	809-453-5
1510	Ocean Paradise	Paul	Gardner	PO Box 8745		Kailua-Kona	HI	94756	US	808-555-8231	808-555-8
1513	Fantastique Aquatica	Susan	Wong	Z32 999 #12A-77 A.A.		Bogota			Columbia	057-1-773434	057-1-773
1551	Marmot Divers Club	Joyce	Marsh	872 Queen St.		Kitchener	Ontario	G3N 2E1	Canada	416-698-0399	426-698-0
1560	The Depth Charge	Sam	Witherspoon	15243 Underwater Fwy.		Marathon	FL	35003	US	800-555-3798	800-555-0
1563	Blue Sports	Theresa	Kunec	203 12th Ave. Box 746		Giribaldi	OR	91187	US	610-772-6704	610-772-6
1624	Makai SCUBA Club	Donna	Siaus	PO Box 8534		Kailua-Kona	HI	94756	US	317-649-9098	317-649-6
1645	Action Club	Michael	Spurling	PO Box 5451-F		Sarasota	FL	32274	US	813-870-0239	813-870-0
1651	Jamaica SCUBA Centre	Barbara	Harvey	PO Box 68		Negril	Jamaica		West Indies	011-3-697043	011-3-697
1680	Island Finders	Desmond	Ortega	6133 1/3 Stone Avenue		St Simons Isle	GA	32521	US	713-423-5675	713-423-5
1984	Adventure Undersea	Gloria	Gonzales	PO Box 744		Belize City			Belize	011-34-09054	011-34-09
2118	Blue Sports Club	Harry	Bathbone	63365 Nez Perce Street		Largo	FL	34684	US	612-897-0342	612-897-0
2135	Frank's Divers Supply	Lloyd	Fellows	1455 North 44th St.		Eugene	OR	90427	US	503-555-2778	503-555-2
2156	Davy Jones' Locker	Tanya	Wagner	246 South 16th Place		Vancouver	BC	K8V 9P1	Canada	803-509-0112	803-509-0

« First Page « < Prev Page < > Next Page > » Last Page »

One more way: ibm_db2's cursor option

```
$startingRow = 20;
$endingRow = 29;
$stmt = db2_exec($stmt, array('cursor' => DB2_SCROLLABLE));
$currentRow = $startingRow;
while (($currentRow <= $endingRow) &&
           $row = db2_fetch_array($stmt, $currentRow)) {
    print "$row[0]\n";
    $currentRow++;
}
```

Note additional param for row number:

array db2_fetch_array (resource $stmt [, int $row_number = -1])

This technique is probably simplest but depends on middleware choice

Performance tips

Performance tips

- **Use persistent connections (see next slides)**
- **Optimize SQL**
  - A few specific tips (there are so many ways):
    - Avoid scalar functions (such as UPPER) in WHERE clause
    - OPTIMIZE FOR n ROWS (when know how many rows)
    - Check out IBM Rochester’s DB2 SQL performance class
- **Index properly**
  - Refer to Plan Cache and Index Advisor
  - Try Encoded Vector Indexes when appropriate
- **System configuration**
  - More memory, more memory!
  - Consider separate memory pool to keep data together

DB2 query optimization

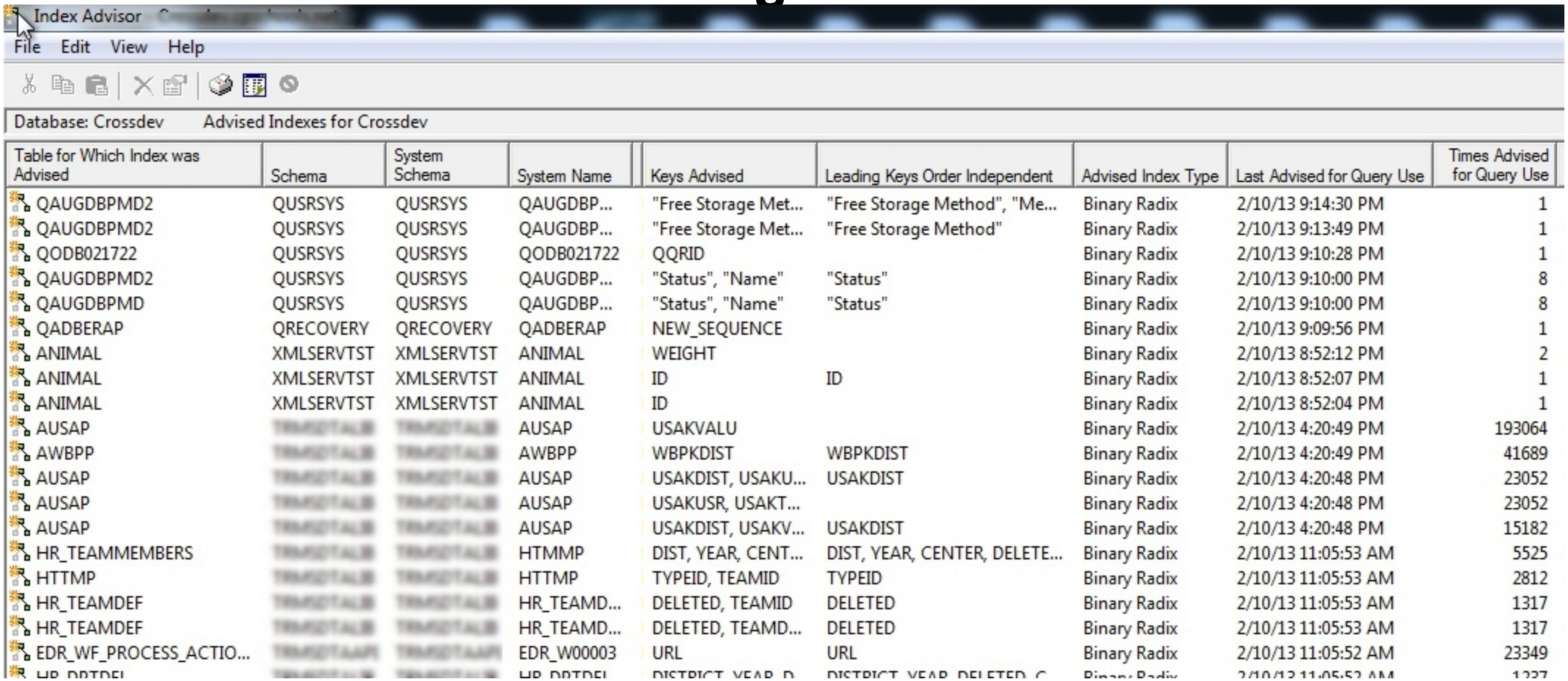
- **IBM i has great tools**

- **I’ll share a couple of favorites**
  - Index Advisor
  - SQL Plan Cache

- **See IBM’s book**
  - IBM i Database Performance and Query Optimization
  - http://pic.dhe.ibm.com/infocenter/iseries/v7r1m0/topic/rzajq/rzajq.pdf

Index Advisor

- **Recommends indexes across all queries**
- **Now in web-based Navigator as well as thick client**

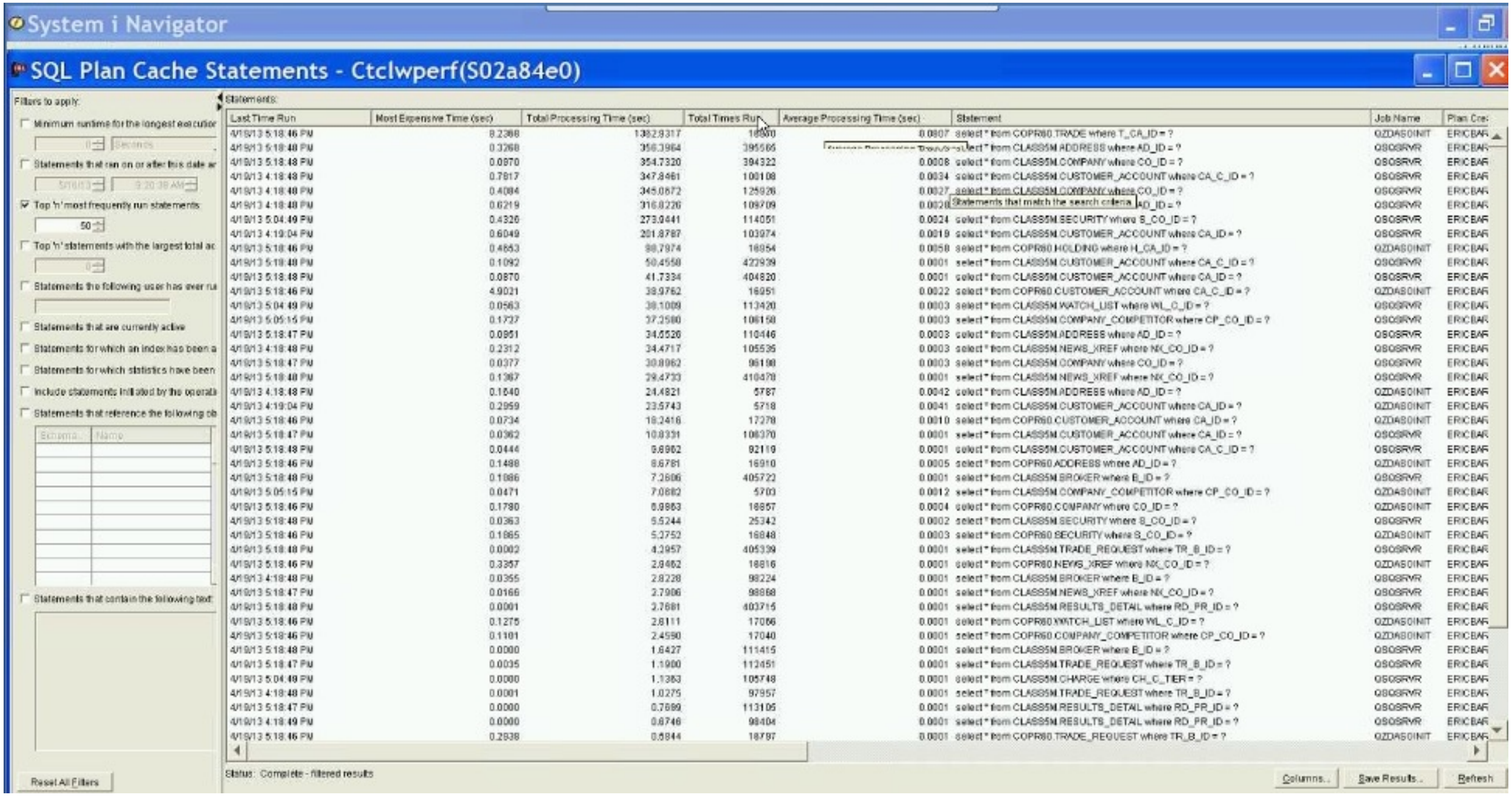


Database: Crossdev Advised Indexes for Crossdev

Table for Which Index was Advised	Schema	System Schema	System Name	Keys Advised	Leading Keys Order Independent	Advised Index Type	Last Advised for Query Use	Times Advised for Query Use
QAUGDBPMD2	QUSRSYS	QUSRSYS	QAUGDBP...	"Free Storage Met...	"Free Storage Method", "Me...	Binary Radix	2/10/13 9:14:30 PM	1
QAUGDBPMD2	QUSRSYS	QUSRSYS	QAUGDBP...	"Free Storage Met...	"Free Storage Method"	Binary Radix	2/10/13 9:13:49 PM	1
QODB021722	QUSRSYS	QUSRSYS	QODB021722	QQRID		Binary Radix	2/10/13 9:10:28 PM	1
QAUGDBPMD2	QUSRSYS	QUSRSYS	QAUGDBP...	"Status", "Name"	"Status"	Binary Radix	2/10/13 9:10:00 PM	8
QAUGDBPMD	QUSRSYS	QUSRSYS	QAUGDBP...	"Status", "Name"	"Status"	Binary Radix	2/10/13 9:10:00 PM	8
QADBERAP	QRECOVERY	QRECOVERY	QADBERAP	NEW_SEQUENCE		Binary Radix	2/10/13 9:09:56 PM	1
ANIMAL	XMLSERVTST	XMLSERVTST	ANIMAL	WEIGHT		Binary Radix	2/10/13 8:52:12 PM	2
ANIMAL	XMLSERVTST	XMLSERVTST	ANIMAL	ID	ID	Binary Radix	2/10/13 8:52:07 PM	1
ANIMAL	XMLSERVTST	XMLSERVTST	ANIMAL	ID		Binary Radix	2/10/13 8:52:04 PM	1
AUSAP	[redacted]	[redacted]	AUSAP	USAKVALU		Binary Radix	2/10/13 4:20:49 PM	193064
AWBPP	[redacted]	[redacted]	AWBPP	WBPKDIST	WBPKDIST	Binary Radix	2/10/13 4:20:49 PM	41689
AUSAP	[redacted]	[redacted]	AUSAP	USAKDIST, USAKU...	USAKDIST	Binary Radix	2/10/13 4:20:48 PM	23052
AUSAP	[redacted]	[redacted]	AUSAP	USAKUSR, USAKT...		Binary Radix	2/10/13 4:20:48 PM	23052
AUSAP	[redacted]	[redacted]	AUSAP	USAKDIST, USAKV...	USAKDIST	Binary Radix	2/10/13 4:20:48 PM	15182
HR_TEAMMEMBERS	[redacted]	[redacted]	HTMMP	DIST, YEAR, CENT...	DIST, YEAR, CENTER, DELETE...	Binary Radix	2/10/13 11:05:53 AM	5525
HTTMP	[redacted]	[redacted]	HTTMP	TYPEID, TEAMID	TYPEID	Binary Radix	2/10/13 11:05:53 AM	2812
HR_TEAMDEF	[redacted]	[redacted]	HR_TEAMD...	DELETED, TEAMID	DELETED	Binary Radix	2/10/13 11:05:53 AM	1317
HR_TEAMDEF	[redacted]	[redacted]	HR_TEAMD...	DELETED, TEAMD...	DELETED	Binary Radix	2/10/13 11:05:53 AM	1317
EDR_WF_PROCESS_ACTIO...	[redacted]	[redacted]	EDR_W00003	URL	URL	Binary Radix	2/10/13 11:05:52 AM	23349
HR_DPTDEF	[redacted]	[redacted]	HR_DPTDEF	DISTRICT, YEAR, D...	DISTRICT, YEAR, DELETED, C...	Binary Radix	2/10/13 11:05:52 AM	1237

SQL Plan Cache

- **Lets you see what queries are REALLY running, who's running them, and how long they take**



SQL Plan Cache Statements - Ctclwperf(S02a84e0)

Filters to apply:

☐ Minimum runtime for the longest executior
0 Seconds

☐ Statements that ran on or after this date ar
5/16/13 9:20:38 AM

☑ Top 'n' most frequently run statements:
50

☐ Top 'n' statements with the largest total ac
0

☐ Statements the following user has ever ru

☐ Statements that are currently active

☐ Statements for which an index has been a

☐ Statements for which statistics have been

☐ Include statements initiated by the operati

☐ Statements that reference the following ob

Schema	Name

☐ Statements that contain the following text:

Statements:

Last Time Run	Most Expensive Time (sec)	Total Processing Time (sec)	Total Times Run	Average Processing Time (sec)	Statement	Job Name	Plan Cre
4/19/13 5:18:46 PM	8.2388	1382.9317	[illegible]	0.0807	select * from COPR80.TRADE where T_CA_ID = ?	QZDASOINIT	ERICBAR
4/19/13 5:18:48 PM	0.3268	356.3964	395565	[illegible]	select * from CLASS5M.ADDRESS where AD_ID = ?	QSQSRVR	ERICBAR
4/19/13 5:18:48 PM	0.0870	354.7320	394322	0.0008	select * from CLASS5M.COMPANY where CO_ID = ?	QSQSRVR	ERICBAR
4/19/13 4:18:48 PM	0.7817	347.8461	100108	0.0034	select * from CLASS5M.CUSTOMER_ACCOUNT where CA_C_ID = ?	QSQSRVR	ERICBAR
4/19/13 4:18:48 PM	0.4084	345.0872	125926	0.0027	select * from CLASS5M.COMPANY where CO_ID = ?	QSQSRVR	ERICBAR
4/19/13 4:18:48 PM	0.6219	316.8226	109709	0.0029	Statements that match the search criteria AD_ID = ?	QSQSRVR	ERICBAR
4/19/13 5:04:49 PM	0.4326	273.9441	114051	0.0024	select * from CLASS5M.SECURITY where S_CO_ID = ?	QSQSRVR	ERICBAR
4/19/13 4:19:04 PM	0.6049	201.8787	103974	0.0019	select * from CLASS5M.CUSTOMER_ACCOUNT where CA_ID = ?	QSQSRVR	ERICBAR
4/19/13 5:18:46 PM	0.4853	98.7974	16954	0.0058	select * from COPR80.HOLDING where H_CA_ID = ?	QZDASOINIT	ERICBAR
4/19/13 5:18:48 PM	0.1092	50.4558	422939	0.0001	select * from CLASS5M.CUSTOMER_ACCOUNT where CA_C_ID = ?	QSQSRVR	ERICBAR
4/19/13 5:18:48 PM	0.0870	41.7334	404820	0.0001	select * from CLASS5M.CUSTOMER_ACCOUNT where CA_ID = ?	QSQSRVR	ERICBAR
4/19/13 5:18:46 PM	4.9021	38.9762	16951	0.0022	select * from COPR80.CUSTOMER_ACCOUNT where CA_C_ID = ?	QZDASOINIT	ERICBAR
4/19/13 5:04:49 PM	0.0563	38.1009	113420	0.0003	select * from CLASS5M.WATCH_LIST where WL_C_ID = ?	QSQSRVR	ERICBAR
4/19/13 5:05:15 PM	0.1727	37.2580	106158	0.0003	select * from CLASS5M.COMPANY_COMPETITOR where CP_CO_ID = ?	QSQSRVR	ERICBAR
4/19/13 5:18:47 PM	0.0951	34.6526	110446	0.0003	select * from CLASS5M.ADDRESS where AD_ID = ?	QSQSRVR	ERICBAR
4/19/13 4:18:48 PM	0.2312	34.4717	105535	0.0003	select * from CLASS5M.NEWS_XREF where NX_CO_ID = ?	QSQSRVR	ERICBAR
4/19/13 5:18:47 PM	0.0377	30.8962	96198	0.0003	select * from CLASS5M.COMPANY where CO_ID = ?	QSQSRVR	ERICBAR
4/19/13 5:18:48 PM	0.1367	29.4733	410478	0.0001	select * from CLASS5M.NEWS_XREF where NX_CO_ID = ?	QSQSRVR	ERICBAR
4/19/13 4:18:48 PM	0.1640	24.4821	5787	0.0042	select * from CLASS5M.ADDRESS where AD_ID = ?	QZDASOINIT	ERICBAR
4/19/13 4:19:04 PM	0.2959	23.5743	5718	0.0041	select * from CLASS5M.CUSTOMER_ACCOUNT where CA_ID = ?	QZDASOINIT	ERICBAR
4/19/13 5:18:46 PM	0.0734	18.2416	17278	0.0010	select * from COPR80.CUSTOMER_ACCOUNT where CA_ID = ?	QZDASOINIT	ERICBAR
4/19/13 5:18:47 PM	0.0362	10.8331	108370	0.0001	select * from CLASS5M.CUSTOMER_ACCOUNT where CA_ID = ?	QSQSRVR	ERICBAR
4/19/13 5:18:48 PM	0.0444	9.8962	92119	0.0001	select * from CLASS5M.CUSTOMER_ACCOUNT where CA_C_ID = ?	QSQSRVR	ERICBAR
4/19/13 5:18:46 PM	0.1488	8.6781	16910	0.0005	select * from COPR80.ADDRESS where AD_ID = ?	QZDASOINIT	ERICBAR
4/19/13 5:18:48 PM	0.1086	7.2606	405722	0.0001	select * from CLASS5M.BROKER where B_ID = ?	QSQSRVR	ERICBAR
4/19/13 5:05:15 PM	0.0471	7.0882	5703	0.0012	select * from CLASS5M.COMPANY_COMPETITOR where CP_CO_ID = ?	QZDASOINIT	ERICBAR
4/19/13 5:18:46 PM	0.1780	6.9863	16857	0.0004	select * from COPR80.COMPANY where CO_ID = ?	QZDASOINIT	ERICBAR
4/19/13 5:18:48 PM	0.0363	5.5244	25342	0.0002	select * from CLASS5M.SECURITY where S_CO_ID = ?	QSQSRVR	ERICBAR
4/19/13 5:18:46 PM	0.1865	5.2752	16848	0.0003	select * from COPR80.SECURITY where S_CO_ID = ?	QZDASOINIT	ERICBAR
4/19/13 5:18:48 PM	0.0002	4.2957	405339	0.0001	select * from CLASS5M.TRADE_REQUEST where TR_B_ID = ?	QSQSRVR	ERICBAR
4/19/13 5:18:46 PM	0.3357	2.9462	16816	0.0001	select * from COPR80.NEWS_XREF where NX_CO_ID = ?	QZDASOINIT	ERICBAR
4/19/13 4:18:48 PM	0.0355	2.8228	98224	0.0001	select * from CLASS5M.BROKER where B_ID = ?	QSQSRVR	ERICBAR
4/19/13 5:18:47 PM	0.0166	2.7906	98868	0.0001	select * from CLASS5M.NEWS_XREF where NX_CO_ID = ?	QSQSRVR	ERICBAR
4/19/13 5:18:48 PM	0.0001	2.7681	403715	0.0001	select * from CLASS5M.RESULTS_DETAIL where RD_PR_ID = ?	QSQSRVR	ERICBAR
4/19/13 5:18:46 PM	0.1275	2.6111	17066	0.0001	select * from COPR80.WATCH_LIST where WL_C_ID = ?	QZDASOINIT	ERICBAR
4/19/13 5:18:46 PM	0.1101	2.4590	17040	0.0001	select * from COPR80.COMPANY_COMPETITOR where CP_CO_ID = ?	QZDASOINIT	ERICBAR
4/19/13 5:18:48 PM	0.0000	1.6427	111415	0.0001	select * from CLASS5M.BROKER where B_ID = ?	QSQSRVR	ERICBAR
4/19/13 5:18:47 PM	0.0035	1.1900	112451	0.0001	select * from CLASS5M.TRADE_REQUEST where TR_B_ID = ?	QSQSRVR	ERICBAR
4/19/13 5:04:49 PM	0.0000	1.1363	105748	0.0001	select * from CLASS5M.CHARGE where CH_C_TIER = ?	QSQSRVR	ERICBAR
4/19/13 4:18:48 PM	0.0001	1.0275	97957	0.0001	select * from CLASS5M.TRADE_REQUEST where TR_B_ID = ?	QSQSRVR	ERICBAR
4/19/13 5:18:47 PM	0.0000	0.7689	113105	0.0001	select * from CLASS5M.RESULTS_DETAIL where RD_PR_ID = ?	QSQSRVR	ERICBAR
4/19/13 4:18:49 PM	0.0000	0.6746	98404	0.0001	select * from CLASS5M.RESULTS_DETAIL where RD_PR_ID = ?	QSQSRVR	ERICBAR
4/19/13 5:18:46 PM	0.2938	0.5844	18787	0.0001	select * from COPR80.TRADE_REQUEST where TR_B_ID = ?	QZDASOINIT	ERICBAR

Status: Complete - filtered results

Bonus tip for query profiling

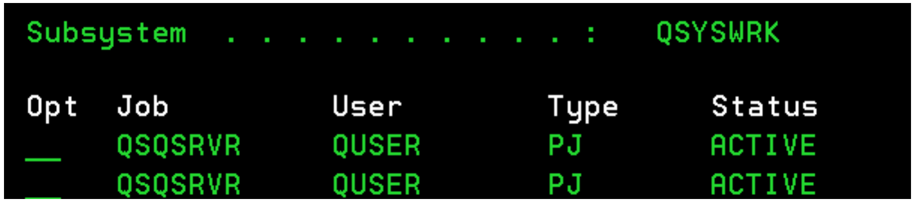
- **Append an SQL comment, containing script name, to your queries**
- **You’ll then be able to tie SQL to the script that generated or ran it**

```
$sql = 'select * from mytable where inv = 45';
// add SQL comment to make SQL easier to track
$sql .= ' -- This query generated by ' .
__METHOD__ . ‘(): ' . __FILE__;
```

- **Logged as:** `select * from mytable where inv = 45`
  `-- This query generated by`
  `Invoice::invoiceQuery(), /www/zendsvr6/htdocs/`
  `ap/payables.php`

Persistent connections

- **Create a pool of database jobs**
  - Known as a “connection pool”
  - You will connect quickly because a job is waiting for you
  - DB2 will...
    - Choose a QSQSRVR job when your PHP job first connects
    - Create new jobs to handle high workload



```
Subsystem . . . . . . . . . . :   QSYSWRK

Opt  Job          User         Type     Status
___  QSQSRVR      QUSER        PJ       ACTIVE
___  QSQSRVR      QUSER        PJ       ACTIVE
```

- **The word “persistent” may be misleading**
  - No guarantee that a browser session reconnects to same job
  - Between requests, cannot rely on maintaining state (QTEMP, library lists). OK within a request, though

db2_pconnect() to connect persistently

- `resource db2_pconnect ( string $database , string $username , string $password [, array $options ] )`

- **Persistent is much faster than non-persistent**
  - db2_pconnect can reuse connections, reducing the time needed to connect (after the first time) to almost zero
  - SQL statement objects can also be reused, speeding queries

- **How db2_pconnect() reuses connections**
  - Connections defined by *database*, *username*, and *password*
  - Tries to reuse an existing connection matching these 3 params
  - db2_close() on a persistent connection does nothing
  - db2_pclose() forces the conn to close

Windows/Linux
dev clients

“How do I develop on non-i, deploy on i?”

- **Developer goal:**
  - Develop IBM i-based PHP code on a non-i machine (Linux, Windows)
  - Connect to DB2 on IBM i using code that will also work on ‘i’ later in production

- **Requires a separate product from IBM**

IBM DB2 Connect from Windows/Linux

- **Purchase IBM's "DB2 Connect"**
  - http://www-01.ibm.com/software/data/db2/db2connect
  - Not free. We'll see how this develops (no more Personal Edition)
- **Your non-i computer will host a local "dummy" DB2 database that actually accesses DB2 on your IBM i**
  - Use the same ibm_db2 functions that you normally do
  - Configuration tips: http://174.79.32.155/wiki/index.php/Tier2/DB2Connect
- **Library lists (system naming) available in DB2 Connect 10.5**

Alternative to DB2 Connect

- ODBC no charge
- ODBC optimized for IBM i, not actively maintained/ supported
- Some have had success with it
- I don’t recommend for mission-critical systems but you are welcome to try it, especially if your software already supports ODBC
- Less portable than DB2 Connect in moving app to/ from IBM i

Logical files

Use ORDER BY and WHERE

- **Good news: SQL and DB2 can use the indexes of DDS-based logical files**

- **Caution: specifying a LF name doesn’t guarantee that its index will be used**

- **Solution: use “ORDER BY” and “WHERE” clauses**
  - Indexes from LFs will be used automatically as appropriate
  - Avoid confusion by specifying only physical files

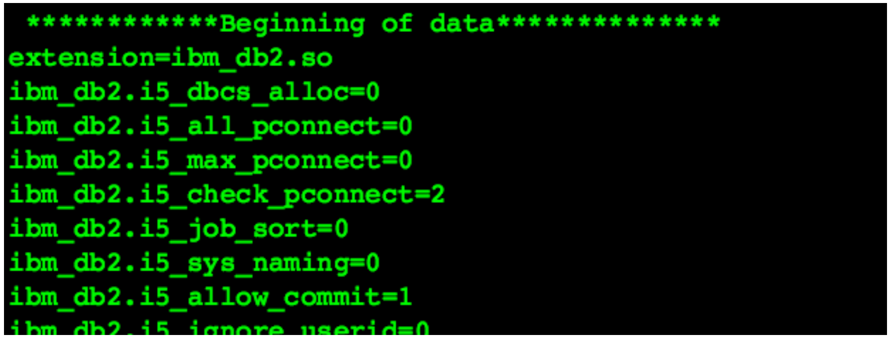
Before 7.1, avoid select/omit in LFs

- Select/omit criteria in LFs (before 7.1) caused the query optimizer to revert to Classic Query Engine (CQE) instead of the SQL Query Engine (SQE)
- SQE is better and newer
- Solution: use a WHERE clause, when possible, instead of select/omit (or upgrade to 7.1/7.2)

New
Global settings

Config file(s) for ibm_db2

- **ibm_db2.ini**
  - Main location for these settings
  - /usr/local/zendsvr6/etc/conf.d/ibm_db2.ini
  - A small file containing only ibm_db2 settings
  - Initial contents:
    `extension=ibm_db2.so`



```
 ************Beginning of data**************
extension=ibm_db2.so
ibm_db2.i5_dbcs_alloc=0
ibm_db2.i5_all_pconnect=0
ibm_db2.i5_max_pconnect=0
ibm_db2.i5_check_pconnect=2
ibm_db2.i5_job_sort=0
ibm_db2.i5_sys_naming=0
ibm_db2.i5_allow_commit=1
ibm_db2.i5_ignore_userid=0
```

- **php.ini**
  - Less common location
  - /usr/local/zendsvr6/etc/php.ini
  - Large file containing hundreds of settings
  - Add or modify settings under the section `[ibm_db2]`

New ibm_db2 config options

- In v1.9.6-sg25 of ibm_db2 shipped with Zend Server 7+
- See change log for news
  - http://174.79.32.155/wiki/index.php/XMLSERVICE/PHPDB2ChangeLog
- Highlights coming right up

Highlights new (slide 1 of 2)

Setting	Default	What it does
ibm_db2.**i5_sys_naming**	**0**,1	1 enables library lists by default (even in LUW DB2 Connect 10.5)
ibm_db2.**i5_blank_userid**	0,**1**	When 0, blank user id/ password become invalid
ibm_db2.**i5_max_pconnect**	**0**-n	Cleans up persistent QSQSRVR jobs every so many connections
ibm_db2.**i5_check_pconnect**	**0**-4	Checks pconnect job. 0 means no test; 1-4 progressively more robust to test for a valid connection. If connection invalid, reconnect fresh

Highlights new (slide 2 of 2)

Setting	Values	What it does
ibm_db2.**i5_log_verbose**	**0**,1	Log DB2 errors in /usr/local/zendsvr(6)/var/logs/php.log **Alan recommends 1**
ibm_db2. **i5_servermode_subsystem**	**[empty]**, *SAME, subsystem (QSYSWRK)	Exposes connection attribute SQL_ATTR_SERVERMODE_SUBSYSTEM to specify subsystem for QSQSRVR jobs
ibm_db2.**i5_guard_profile**	**0**,1	Restore job's original "current user" at end of PHP request (this feature subject to change)

Questions

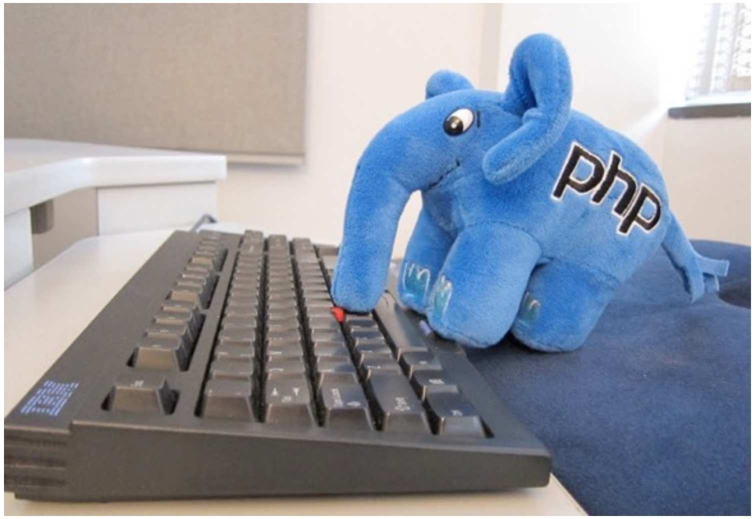
Contact and tips

Alan Seiden

Alan Seiden Consulting

Ho-Ho-Kus, NJ

Free newsletter:
http://alanseiden.com/tips



alan@alanseiden.com • 201-447-2437 • twitter: @alanseiden